

日本国特許庁  
JAPAN PATENT OFFICE

01 045

RPS

JC978 U.S. PTO  
10/085328



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2001年 3月 8日

出願番号

Application Number:

特願2001-064713

出願人

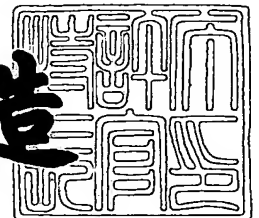
Applicant(s):

インターナショナル・ビジネス・マシーンス・コーポレーション

2001年11月30日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3105418

【書類名】 特許願  
【整理番号】 JP9010045  
【あて先】 特許庁長官殿  
【国際特許分類】 H04N 1/40  
G06F 15/66

【発明者】

【住所又は居所】 滋賀県野洲郡野洲町大字市三宅 8 0 0 番地 日本アイ・  
ビー・エム株式会社 野洲事業所内

【氏名】 福島 好伸

【発明者】

【住所又は居所】 滋賀県野洲郡野洲町大字市三宅 8 0 0 番地 日本アイ・  
ビー・エム株式会社 野洲事業所内

【氏名】 北川 雅彦

【発明者】

【住所又は居所】 滋賀県野洲郡野洲町大字市三宅 8 0 0 番地 日本アイ・  
ビー・エム株式会社 野洲事業所内

【氏名】 中野 宏毅

【特許出願人】

【識別番号】 390009531

【住所又は居所】 アメリカ合衆国 1 0 5 0 4、ニューヨーク州アーモンク  
(番地なし)

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレ  
ーション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【電話番号】 0462-73-3318

【代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【代理人】

【識別番号】 100106699

【弁理士】

【氏名又は名称】 渡辺 弘道

【復代理人】

【識別番号】 100094248

【弁理士】

【氏名又は名称】 楠本 高義

【手数料の表示】

【予納台帳番号】 012922

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【包括委任状番号】 0004480

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ補正装置及びデータ補正方法

【特許請求の範囲】

【請求項 1】 複数の入力素子で構成された入力部が取り込んだ入力データを補正するデータ補正装置において、

前記入力データの補正を行うための高次多項式を、直交展開するための展開係数及び基底関数のそれぞれの演算データである展開係数配列データ及び基底関数配列データが記憶された演算用メモリと、

前記展開係数配列データ及び基底関数配列データを利用して入力データの補正を行う演算部と、

を含むデータ補正装置。

【請求項 2】 前記演算用メモリ及び演算部が複数個である請求項 1 に記載のデータ補正装置。

【請求項 3】 前記展開係数配列データが、前記複数の入力素子ごとに、前記展開係数を演算して求められたデータである請求項 1 または 2 に記載のデータ補正装置。

【請求項 4】 前記基底関数配列データが、前記入力データが取りうる値ごとに、前記基底関数を演算して求められたデータである請求項 1 乃至 3 に記載のデータ補正装置。

【請求項 5】 前記基底関数が、チェビシェフ関数である請求項 1 乃至 4 に記載のデータ補正装置。

【請求項 6】 前記チェビシェフ関数が 2 次乃至 4 次のチェビシェフ関数である請求項 5 に記載のデータ補正装置。

【請求項 7】 前記高次多項式が、LUT (Look up table) 変換を近似した式である請求項 1 乃至 6 に記載のデータ補正装置。

【請求項 8】 前記演算部がDSP (Digital signal processor) の演算回路であり、前記演算用メモリが該DSPの内部メモリである請求項 1 乃至 7 に記載のデータ補正装置。

【請求項 9】 複数の入力素子で構成された入力部が取り込んだ入力データを、

演算部が補正するデータ補正方法において、

前記入力データの補正を行うための高次多項式を、直交展開するための展開係数及び基底関数のそれぞれの演算データである演算用メモリに記憶された展開係数配列データ及び基底関数配列データを、前記演算部が得るステップと、

前記演算部が、前記展開係数配列データと前記基底関数配列データを乗算するステップと、

前記乗算するステップによって求められた結果を、前記演算部が加算するステップと、

を含むデータ補正方法。

【請求項 1 0】 前記演算部が得るステップは、前記演算部が前記入力データ及び複数の入力素子を識別する画素番号を参照して行われる請求項 9 に記載のデータ補正方法。

【請求項 1 1】 前記演算部は、前記入力データを参照することによって、前記基底関数配列データを得る請求項 1 0 に記載のデータ補正方法。

【請求項 1 2】 前記演算部は、前記複数の入力素子を識別する画素番号を参照することによって、前記展開係数配列データを得る請求項 1 0 または 1 1 に記載のデータ補正方法。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、CCD (charge coupled device) から出力されたデータを補正するためのデータ補正装置及びデータ補正方法に関する。

【 0 0 0 2 】

【従来の技術】

従来、入力部としてラインセンサ、平面センサ及びデジタルカメラなどを利用した画像処理を行う場合、ラインセンサなどを構成する各 CCD は、受光感度にばらつきがある。従って、図 5 に示す CCD の入出力特性は、CCD ごとに異なる。CCD の入出力特性を補正するために、LUT (Look up table) が必要になる。LUT を使用した LUT 変換は、ハードウェアあるいはソフトウェアに

よって実行されていた。

【0003】

ハードウェアでLUT変換を実行する場合、入力データをSRAM (Static RAM) のアドレス線に入力し、SRAMに記憶された補正後のデータを取得し、このデータを出力データとする。LUT変換の例として、8bitのデータ（取りうる値は0-255）の変換の場合、予め変換後の数値を格納した要素数が256個の配列をメモリに用意し、入力データの値を引数として配列にアクセスする。アクセスした配列のデータがLUT変換後のデータである。配列のメモリ使用量は256バイトとなる。10bitのデータの場合（とりうる値は0-1023）は、要素数が1024個の配列を用意する。メモリは通常8ビット単位で構成されるので、配列のメモリ使用量は2048バイトとなる。

【0004】

例えばCCDの画素数が3000の場合、CCDの画素ごとにLUTが必要となるので、LUTのメモリ使用量は、 $3000 \times 2KB = 6MB$ となる。しかし、LUT変換は、ラインセンサの画像出力と同等の速度（一例としてピクセルレート30MHz）で動作しなければならず、6MBもの高速SRAMで構成する必要がある。これは、价格的、実装面積的に制約が大きく、実現が困難である。

【0005】

また、デジタルデータの処理を行うためのマイクロプロセッサであるDSP (Digital signal processor) を使用し、ソフトウェアによりLUT変換を構成することも可能である。一例として、30個のDSPで3000画素を分割して処理する場合、各DSPは100画素を処理する。従って、各DSPにはLUT変換のために200KBの内部SRAMが必要となる。

【0006】

しかし、DSPの内部SRAMは大きくても48KB程度までであるため、現実的ではない。更に、DSPの本来の目的は画像データ取得後の画像処理にあるため、内部SRAMの一部に大容量のLUTが常駐することは好ましくない。

【0007】

ソフトウェアによって行うLUT変換の効率を改善するため、LUT変換を多

項式で近似して行う方法もある。図5に示すようにCCDごとの入出力特性は非線形である。入出力特性は連続であり、例えば4次関数を使用して近似する場合、入出力特性カーブは式1のようになる。係数は、予め入出力特性を基に決定しておく。

【0008】

【式1】

$$f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \quad (1)$$

【0009】

式1の演算を行うためには、乗算10回、加算4回が必要である。ただし、倍精度演算が必要であり、単精度演算に換算すると、乗算40回に相当する。なお、加算回数についてはプロセッサ・アーキテクチャによって変動するので、比較を行わない。ピクセルレートを30MHzとすると、DSP1つ当たりの乗算の処理能力は、少なくとも1200メガ演算/秒の処理能力が必要である。DSPの現実的な処理能力はDSPを制御するプログラムによって異なる。しかし、1000メガ演算/秒を超えており、更に加算を行う必要があるため、DSPの処理能力をはるかに超えている。

【0010】

また、式1を変形することによって、演算回数を減らす場合、例えば式2のように変形する。

【0011】

【式2】

$$f(x) = (((a_4x + a_3)x + a_2)x + a_1)x + a_0 \quad (2)$$

【0012】

式2の場合、乗算4回、加算4回で済む。しかし、上記と同じように単精度演算に換算すると、乗算16回になる。ピクセルレートを30MHzとすると、DSP1つ当たり少なくとも480メガ演算/秒の乗算の処理能力が必要である。更に加算が必要であるため、1000メガ演算/秒を越える恐れがあり、DSP

の処理能力を超えてしまう場合がある。

【0013】

【発明が解決しようとする課題】

そこで本発明の目的は、CCDから出力されたデータを処理する際に、メモリ使用量及び演算量を抑えたデータ補正装置及びデータ補正方法を提供することにある。

【0014】

【課題を解決するための手段】

本発明のデータ補正装置の要旨とするところは、複数の入力素子で構成された入力部に入力された入力データを補正するデータ補正装置において、前記データの補正を行うための高次多項式を、直交展開するための展開係数及び基底関数のそれぞれの演算データである展開係数配列データ及び基底関数配列データが記憶された演算用メモリと、前記展開係数配列データ及び基底関数配列データを利用して入力データの補正を行う演算部と、を含むことにある。

【0015】

また、本発明のデータ補正方法の要旨とするところは、複数の入力素子で構成された入力部に入力された入力データを、演算部が補正するデータ補正方法において、前記入力データの補正を行うための高次多項式を、直交展開するための展開係数及び基底関数のそれぞれの演算データである演算用メモリに記憶された展開係数配列データ及び基底関数配列データを、前記演算部が得るステップと、前記演算部が、前記展開係数配列データと前記基底関数配列データを乗算するステップと、前記乗算するステップによって求められた結果を、前記演算部が加算するステップと、を含むことにある。

【0016】

【発明の実施の形態】

本発明のデータ補正装置及びデータ補正方法の実施の形態を図面を基に説明する。

【0017】

図1に示すように本発明のデータ補正装置は、データの補正を行うための高次



多項式を直交展開したときの展開係数及び基底関数を演算した結果である演算データが、それぞれ展開係数配列データ及び基底関数配列データとして記憶された演算用メモリ22と、その展開係数配列データ及び基底関数配列データを参照してデータの補正を行う演算部20と、を含む。展開係数配列データ及び基底関数配列データは、演算用メモリ22の展開係数配列及び基底関数配列を構成する要素である。演算用メモリ22及び演算部20は複数設けられている。また、2次元のデータを取り込み入力データとして入出力制御部16に出力する入力部12、入力データを記憶する複数のデータ用メモリ24、入力部12で取り込まれたデータを、演算部20を介して各データ用メモリ24に振り分ける入出力制御部16、入出力制御部16及び各演算部20を制御する演算制御部14及び演算部20とPCI BUS (Peripheral Component Interconnect BUS) 28の接続を制御するPCIインターフェース26が含まれる。入力部12は複数のCCD (入力素子) で構成されているラインセンサである。本発明において、演算用メモリ22はSRAM (Static RAM) を使用し、データ用メモリ24はSDRAM (Synchronous Dynamic RAM) を使用する。

## 【0018】

演算部20及び演算用メモリ22は、それぞれDSP (Digital signal processor) 18の演算回路及び内部メモリである。入力部12で取り込まれた入力データは1ラインずつ所定の順序でDSP18に分配され、並列処理により処理効率を高めるようにされている。一例として、CCDの数が3000個の場合、DSPの数は30個である。また、1つ当たりのDSPのクロックは250MHzである。入出力制御部16は、FIFO (First-In First-Out) を使用する。データ用メモリ24の容量は48MBである。

## 【0019】

入力部12で取り込まれたデータの補正を行うための高次多項式は、LUT (Look up table) 変換を行うために近似された多項式であり、従来技術で示した式1である。展開計数及び基底関数は、式1を直交展開するとき使用するものである。一般に、区間 $[0, T]$ で定義された関数 $f(x)$ は式3のように級数展開される。

【0020】

【式3】

$$f(x) = \sum_{m=0}^{\infty} b_m \psi_m(x) \quad (3)$$

【0021】

式3の演算結果である出力画素値は、入力データの補正後のデータである。 $\psi_m(x)$ は基底関数、 $b_m$ は展開係数である。 $\psi_m(x)$ が正規直交基底の場合、 $b_m$ は式4のように求められる。

【0022】

【式4】

$$b_m = \int_0^T f(x) j[\psi_m(x)] dx \quad (4)$$

【0023】

$j[\psi_m(x)]$ は、 $\psi_m(x)$ の複素共役である。例えば、CCDが100個あれば、そのそれぞれについて式4を演算して演算データを求め、図3(a)に示すように、この演算データを展開係数配列データ32としてそれぞれの演算用メモリ22に記憶し、展開係数配列30を構成する。図3(a)において、 $A_1$ 乃至 $A_p$ は展開係数配列の配列要素番号(添字)であり、CCDの数が100個であれば $P=100$ になる。即ち、展開係数配列の配列要素番号の数は、CCDの数に対応する。更に、本発明において、展開係数配列30は0乃至4次のそれぞれの展開係数について作成する。即ち、5個の展開係数配列30が演算用メモリ22に構成される。従って、CCDの数が100個の場合、演算用メモリ22に記憶される全ての展開配列データ32は500個になる。また、次元数をM次に変更することによって、0次乃至M次の展開係数配列30を作成することが可能である。

【0024】

$f(x)$ が、以下の式5に示すm次関数で近似される場合、基底関数としてチ

ェビシェフ関数 (Chebychev) を選択することが可能になり、 $m=4$  の場合は、式 6 のように近似関数  $f'(x)$  が得られる。

【0025】

【式 5】

$$f(x) \cong a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0 \quad (5)$$

【0026】

【式 6】

$$f'(x) = \sum_{m=0}^4 b_m C[\psi_m(x)] \quad (6)$$

【0027】

式 6 において  $C[\psi_m]$  はチェビシェフ関数である。0 次乃至 4 次のチェビシェフ関数  $C[\psi_m]$  を以下の式 7 乃至 11 に示す。

【0028】

【式 7】

$$C[\psi_0] = k_0 \quad (7)$$

【0029】

【式 8】

$$C[\psi_1] = k_1 (x - x_c) \quad (8)$$

【0030】

【式 9】

$$C[\psi_2] = k_2 \left( (x - x_c)^2 - \frac{n^2 - 1}{12} \right) \quad (9)$$

【0031】

【式 10】

$$C[\psi_3] = k_3 \left( (x - x_c)^3 - \frac{3n^2 - 7}{20} (x - x_c) \right) \quad (10)$$

【0032】

【式 11】

$$C[\psi_4] = k_4 \left( (x - x_c) - \frac{3n^2 - 13}{14} (x - x_c)^2 + \frac{3(n^2 - 1)(n^2 - 9)}{560} \right) \quad (11)$$

【0033】

式 7 乃至式 11 の  $x_c$  は  $x$  の定義域の中央値、 $n$  は  $x$  を間隔 1 で離散化したときのサンプル数である。定義域は、入力データの取りうる値である。本発明において入力データは、CCD が 2 次元の画像のデータを取り込み、出力した輝度のデータである。入力データが 10 bit のデータの場合、定義域は 0 乃至 1023 になる。 $k_m$  ( $m = 0, 1, 2, 3, 4$ ) は、正規化を行うための係数である。

【0034】

ここで、式 6 は以下のような式 12 になる。

【0035】

【式 12】

出力画素値＝

$$\begin{aligned} & 4 \text{ 次展開係数配列 [画素番号]} \times 4 \text{ 次チェビシェフ関数配列 [入力画素値]} \\ & + 3 \text{ 次展開係数配列 [画素番号]} \times 3 \text{ 次チェビシェフ関数配列 [入力画素値]} \\ & + 2 \text{ 次展開係数配列 [画素番号]} \times 2 \text{ 次チェビシェフ関数配列 [入力画素値]} \\ & + 1 \text{ 次展開係数配列 [画素番号]} \times 1 \text{ 次チェビシェフ関数配列 [入力画素値]} \\ & + 0 \text{ 次展開係数配列 [画素番号]} \times 0 \text{ 次チェビシェフ関数配列 [入力画素値]} \end{aligned} \quad (12)$$

【0036】

式 12 の演算では、乗算 5 回、加算 4 回になっている。更に、演算回数を減ら

すために、式 6 の右辺の 0 次と 1 次の項をまとめて記述すると式 13 になる。

【0037】

【式 13】

$$b_1 k_1 (x - x_c) + b_0 k_0 \quad (13)$$

【0038】

式 13 を変形すると式 14 になる。

【0039】

【式 14】

$$b_1 k_1 x + (b_0 k_0 - b_1 k_1 x_c) \quad (14)$$

【0040】

式 12 の 1 次の展開係数を  $b_1 k_1$ 、0 次の展開係数を  $(b_0 k_0 - b_1 k_1 x_c)$  と定義することによって、0 次及び 1 次のチェビシェフ関数の項は、それぞれ 1 と  $x$  になる。 $x$  は入力データ、即ち、入力画素値であるため、演算部 20 が行う式 12 の演算は、以下の式 15 のようになる。

【0041】

【式 15】

出力画素値＝

4 次展開係数配列 [画素番号] × 4 次チェビシェフ関数配列 [入力画素値]  
 + 3 次展開係数配列 [画素番号] × 3 次チェビシェフ関数配列 [入力画素値]  
 + 2 次展開係数配列 [画素番号] × 2 次チェビシェフ関数配列 [入力画素値]  
 + 1 次展開係数配列 [画素番号] × 入力画素値  
 + 0 次展開係数配列 [画素番号]

(15)

【0042】

式 15 より、式 9 乃至 11 である 2 次乃至 4 次のチェビシェフ関数を全ての定義域で演算して演算データを求め、図 3 (b) に示すように、チェビシェフ関数配列 (基底関数配列) データ 36 としてそれぞれの演算用メモリ 22 に記憶し、チェビシェフ関数配列 (基底関数配列) 34 を構成する。図 3 (b) に示す  $B_1$

乃至  $B_q$  はチェビシェフ関数配列の配列要素番号（添字）であり、入力データが 10 bit の場合、 $q = 1024$  になる。チェビシェフ関数配列の配列要素番号は、入力データの取りうる値と同じ値を使用する。また、2 次乃至 4 次のチェビシェフ関数のそれぞれについて、チェビシェフ関数配列データ 36 を求めるため、3 個のチェビシェフ関数配列 34 が、演算用メモリ 22 に作成される。従って、入力データが 10 bit の場合、 $3 \times 1024 = 3072$  個のチェビシェフ関数配列データ 36 が、演算用メモリ 22 に記憶される。なお、2 次乃至 4 次のチェビシェフ関数についてチェビシェフ関数配列 34 を作成したが、次元数を  $M$  次に変更することによって、チェビシェフ関数配列 34 も 2 次乃至  $M$  次に変更される。

## 【0043】

式 15 の演算結果である出力画素値が、入力データを補正したデータである。演算回数は、乗算が 4 回、加算が 4 回になっている。式 15 より、0 次及び 1 次のチェビシェフ関数の演算結果は、それぞれ 1 と入力画素値になっているため演算用メモリ 22 には記憶されない。

## 【0044】

画素番号は、どの CCD が使用されたかを表すものであり、式 15 において  $m$  次展開係数配列 [画素番号] ( $m$  は 0 乃至 4) は、演算部 20 が画素番号を引数として、演算用メモリ 22 の 0 乃至 4 次のそれぞれの展開係数配列 30 にアクセスした際の展開係数配列データ 32 である。即ち、演算部 20 は画素番号を引数として演算用メモリ 22 の 0 乃至 4 次のそれぞれの展開係数配列 30 にアクセスし、アクセスされた画素番号の展開係数配列データ 32 を得る。

## 【0045】

入力画素値は CCD から出力された入力データであり、 $m$  次チェビシェフ関数配列 [入力画素値] ( $m$  は 2 乃至 4) は、演算部 20 が入力画素値を引数として、演算用メモリ 22 の 2 乃至 4 次のそれぞれのチェビシェフ関数配列 34 にアクセスした際のチェビシェフ関数配列データ 36 である。即ち、演算部 20 は入力画素値を引数として演算用メモリ 22 の 2 次乃至 4 次のそれぞれのチェビシェフ関数配列 34 にアクセスし、アクセスされた入力画素値のチェビシェフ関数配列

データ 36 を得る。

【0046】

また、基底関数であるチェビシェフ関数は、図4に示すように、定義域（0乃至1023）では発散しないことが保証されており、16ビットで表現できる。式15では、2項の16ビット乗算（16ビット×16ビット）であるため、乗算結果は単精度演算である32ビットになる。即ち式15の演算は単精度演算になり、計算時間が短縮される。

【0047】

式15より、5個の展開係数配列30と3個のチェビシェフ関数配列34とが、それぞれの演算用メモリ22に構成される。例えば、CCDの数が100個で、入力データの取りうる値が0乃至1023である場合、1つ当たりの展開係数配列30の展開係数配列データ32は100個、1つ当たりのチェビシェフ関数配列34のチェビシェフ関数配列データ36は1024個になる。従って、500個の展開係数配列データ32と、3072個のチェビシェフ関数配列データ36が演算用メモリ22に記憶される。

【0048】

次に、本発明のデータ補正装置10を使用したデータ補正方法について説明する。データの補正は図2のフローチャートに従って行われる。入力部12であるラインセンサが、1ラインずつ2次元の画像データを読み取り、入力画素データとして入出力制御部16に出力する（ステップ1）。1ラインに100個のCCDがある場合、100個の入力画素データが読み取られる。また、入力画素データを10bitのデータとした場合、取りうる値は0乃至1023の整数である。1ラインの100個の入力データは入出力制御部16によって、演算部20を介して複数あるデータ用メモリ24のいずれかに記憶される。入力データが記憶されるデータ用メモリ24は、番号付けを行っておき、順番に入力データを記憶する。記憶するとき、CCDごとに画素番号を決定し、データ用メモリ24の配列の配列要素番号（添字）を画素番号にすることによって、CCDからの入力データを配列のその画素番号の領域に記憶する。各DSP18が1ラインの入力画素データを順次に受け取る。

## 【0049】

データ用メモリ24に記憶された入力データは、データ用メモリ24に接続された演算部20によって補正される。演算用メモリ22には、LUT変換を近似した高次多項式を直交展開するための展開係数及び基底関数（チェビシェフ関数）を使用して演算された、0乃至4次の展開係数配列データ32及び2次乃至4次のチェビシェフ関数配列データ36が記憶されている。演算部20は、データ用メモリ24に記憶された1ラインの画素の各画素毎に、画素番号を引数として、演算用メモリ22の0次乃至4次のそれぞれの展開係数配列30にアクセスし、それぞれの展開係数配列30から展開係数配列データ32を取得する（ステップ2）。例えば画素番号が $A_1$ の場合、図3（a）において、 $A_1$ を引数として演算用メモリ22の展開係数配列30にアクセスし、その $A_1$ を配列要素番号として記録されている展開係数配列データ32を取得する。なお、図3（a）の展開係数配列30は、0乃至4次について有するので、そのそれぞれについて展開係数配列データ32の取得を行う。

## 【0050】

また、演算部20は、データ用メモリ24に記憶された入力データ（入力画素値）を引数として、2乃至4次のそれぞれのチェビシェフ関数配列34にアクセスし、それぞれのチェビシェフ関数配列34からチェビシェフ関数配列データ36を得る（ステップ3）。例えば入力データが $B_1$ の場合、図3（b）において、 $B_1$ を引数として演算用メモリ22のチェビシェフ関数配列34にアクセスし、その $B_1$ を配列要素番号として記録されているチェビシェフ関数配列データ36を取得する。なお、図3（b）のチェビシェフ関数配列34は、2乃至4次について有するので、そのそれぞれについてチェビシェフ関数配列データ36の取得を行う。

## 【0051】

演算部20は、演算用メモリ22に記憶された0次乃至4次の展開係数配列データ32及び2次乃至4次のチェビシェフ関数配列データ36を取得した後、式15にその展開係数配列データ32及びチェビシェフ関数配列データ36を代入し、演算（乗算及び加算）を行う（ステップ4）。なお、1次展開係数配列デー



タ 3 2 に乗算されるのは入力データ（入力画素値）である。

【 0 0 5 2 】

式 1 5 の演算を行うことによって、入力データを補正した値である出力画素値が求められる。求められた出力画素値は、P C I インターフェース 2 6 を介して P C I B U S 2 8 に出力される。

【 0 0 5 3 】

次に、本発明のデータ補正装置 1 0 を使用した場合のメモリ使用量について説明する。画素数（C C D 数）を  $I_{max}$ 、入力データの取りうる値（定義域）を  $n$ 、高次多項式の次元数を  $M$  とする。展開係数配列及びチェビシェフ関数配列を合わせた配列の要素数（データ数）は、 $I_{max} \times (M + 1) + (M - 1) \times n$  になる。 $I_{max} = 100$ 、 $n = 1024$ 、 $M = 4$  の場合、配列の要素数は 3 5 7 2 になり、メモリの使用量は 7 1 4 4 バイトになる。

【 0 0 5 4 】

従来技術である L U T 変換を使用した場合、配列の要素数は  $I_{max} \times n$  であり、 $I_{max} = 100$ 、 $n = 1024$  の場合、配列の要素数は 1 0 2 4 0 0 になり、メモリの使用量は 2 0 4 8 0 0 バイトになる。また、従来技術の高次多項式（式 2）を完全に計算した場合、配列の要素数は  $I_{max} \times (M + 1)$  であり、 $I_{max} = 100$ 、 $M = 4$  であれば、配列の要素数は 5 0 0 になり、メモリの使用量は 1 0 0 0 バイトになる。

【 0 0 5 5 】

更に、本発明のデータ補正装置 1 0 を使用した場合の演算負荷について説明する。式 1 5 より乗算 4 回、加算 4 回である。なお、倍精度加算を単精度加算の繰り返しで行う場合の繰り返し回数は、プロセッサアーキテクチャーによって変動するので、比較を行わない。演算用メモリ 2 2 に記憶された展開係数配列データ 3 2 及びチェビシェフ関数配列（基底関数配列）データ 3 6 は、正規化されたものである。従って、乗算は単精度乗算（1 6 ビット  $\times$  1 6 ビット）であり、単精度乗算回数は 4 回である。ピクセルレート 3 0 M H z の場合、1 2 0 メガ演算／秒の演算負荷になる。

【 0 0 5 6 】

従来のLUT変換の場合、演算は行われないので演算回数は0回である。また、高次多項式(式2)を完全に演算する場合、乗算4回になる。この場合の乗算は、長精度乗算(32ビット×32ビット)であり、長精度乗算を単精度乗算に換算する場合、たすきがけで4回の単精度乗算を行う必要とするため、長精度乗算4回は単精度乗算16回になる。ピクセルレート30MHzの場合、480メガ演算/秒の演算負荷になる。

【0057】

上記のメモリの使用量及び演算負荷を表1に示す。

【0058】

【表1】

	演算用メモリの使用量 (バイト)	演算負荷 (単精度乗算回数 /CCD)
本発明	7144	4
LUT変換を使用した場合	204800	0
式2を使用した場合	1000	16

【0059】

表1より、本発明のデータ補正装置10及びデータ補正方法によって、LUT変換を使用した場合のように、演算用メモリ22の使用量が必要以上に大きくなることはない。即ち、演算用メモリ22の容量を小さくすることができ、価格を抑えたり、実装面積を小さくできる。また、LUT変換を近似した高次多項式のように、演算部20の処理能力以上の処理速度を必要とされることはない。基底関数を正規化することによって、演算が単精度演算になり、演算スピードが速くなる。基底関数にチェビシェフ関数を選択することにより、0次及び1次のチェビシェフ関数配列データを演算用メモリ22に記憶することはなく、演算用メモリ22の使用量を抑えられる。更に、0次のチェビシェフ関数が1であるため、0次の展開係数配列データと0次のチェビシェフ関数配列データの乗算を行う必要はなく、演算時間を抑えることができる。

【0060】

以上、本発明のデータ補正装置及びデータ補正方法について実施形態を記載したが、本発明のデータ補正装置及びデータ補正方法は上記の実施形態に限定されるものではない。例えば、基底関数は、正規直交系のものであればチェビシェフ関数以外の関数を使用することができる。

## 【 0 0 6 1 】

また、展開係数が無限個数にならないような基底関数を使用することも可能である。

## 【 0 0 6 2 】

求められた出力画素値は、P C I インターフェース 2 6 を介して P C I B U S 2 8 に出力されたが、D S P 1 8 演算部 2 0 が画像認識の機能を備えている場合、出力画素値を使用して画像認識を行うことができる。

## 【 0 0 6 3 】

入力部 1 2 がラインセンサではなく平面センサなどを使用した場合、1 度に 2 次元画像を取り込み、取り込んだ 2 次元画像を 1 ラインずつ出力することもできる。

## 【 0 0 6 4 】

その他、本発明はその趣旨を逸脱しない範囲で当業者の知識に基づき種々なる改良、修正及び変形を加えた態様で実施できるものである。

## 【 0 0 6 5 】

## 【発明の効果】

本発明のデータ補正装置によると、演算用メモリに展開係数配列及び基底関数配列を記憶することにより、演算部での演算時間を抑えることができる。また、演算用メモリの消費量も抑えることができる。

## 【 0 0 6 6 】

また、本発明のデータ補正方法によると、単精度演算回数が乗算 4 回と加算 4 回であり、L U T 変換を近似した高次多項式をそのまま演算するのに比べて、単精度演算回数が少なくなっている。従って、演算スピードを早くすることができる。

## 【図面の簡単な説明】

【図 1】

本発明のデータ補正装置の構成を示す図である。

【図 2】

入力データの補正方法を示すフローチャートである。

【図 3】

演算用メモリ内の配列を示す図あり、(a)は展開係数配列を示す図であり、(b)はチェビシェフ関数配列を示す図である。

【図 4】

チェビシェフ関数を示す図である。

【図 5】

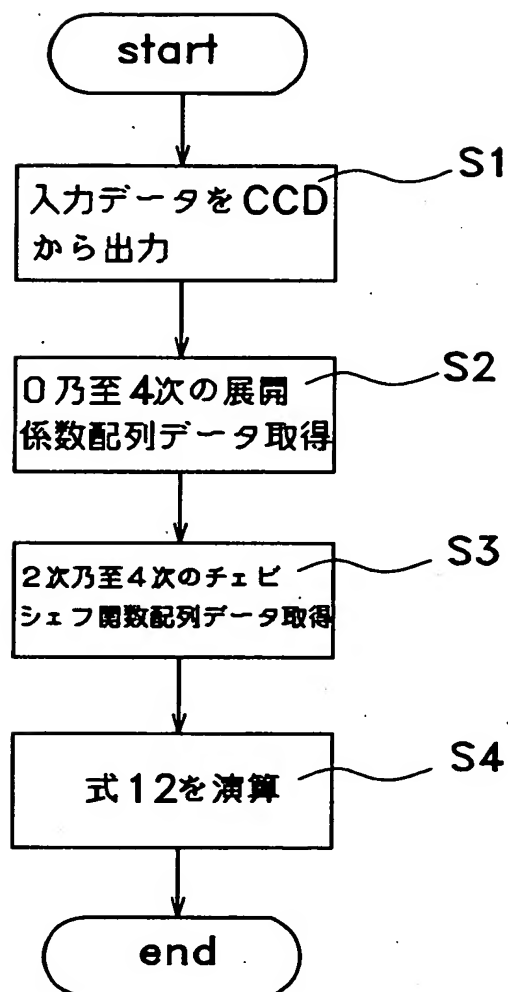
CCDの入出力特性を示す図である。

【符号の説明】

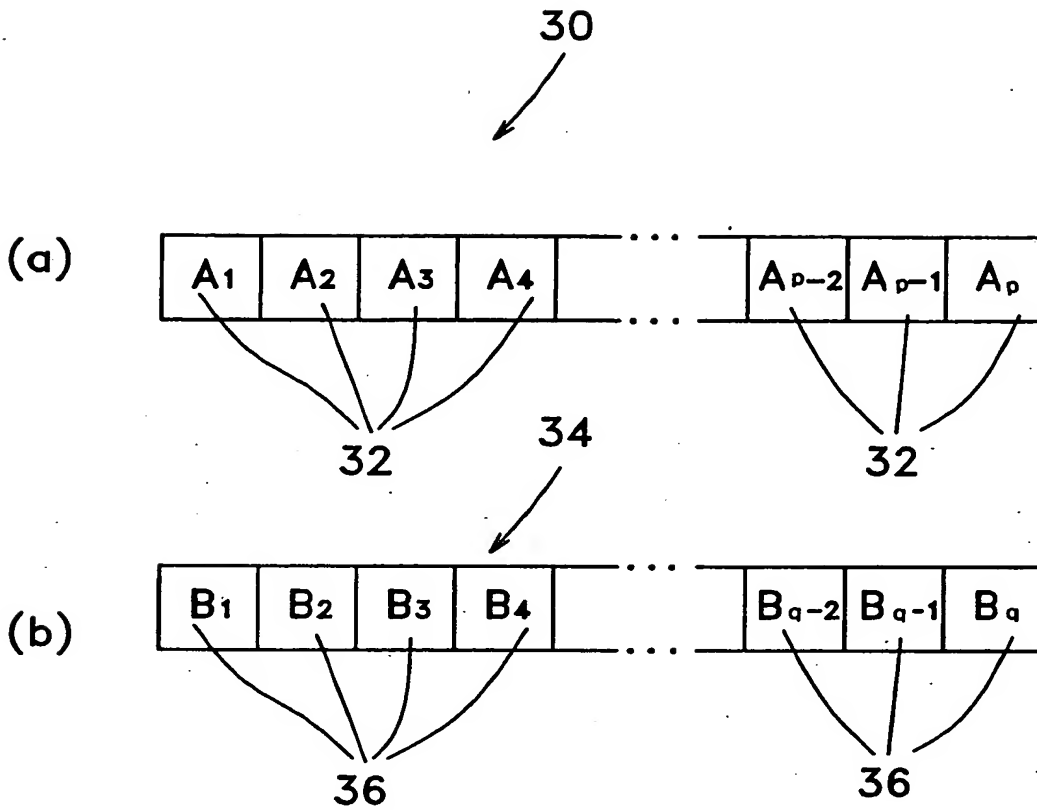
- 10 : データ補正装置
- 12 : ラインセンサ
- 14 : 演算制御部
- 16 : 入出力制御部
- 18 : DSP
- 20 : 演算部
- 22 : 演算用メモリ
- 24 : データ用メモリ
- 26 : PCIインターフェース
- 28 : PCIBUS
- 30 : 展開係数配列
- 32 : 展開係数配列データ
- 34 : チェビシェフ関数配列 (基底関数配列)
- 36 : チェビシェフ関数配列データ



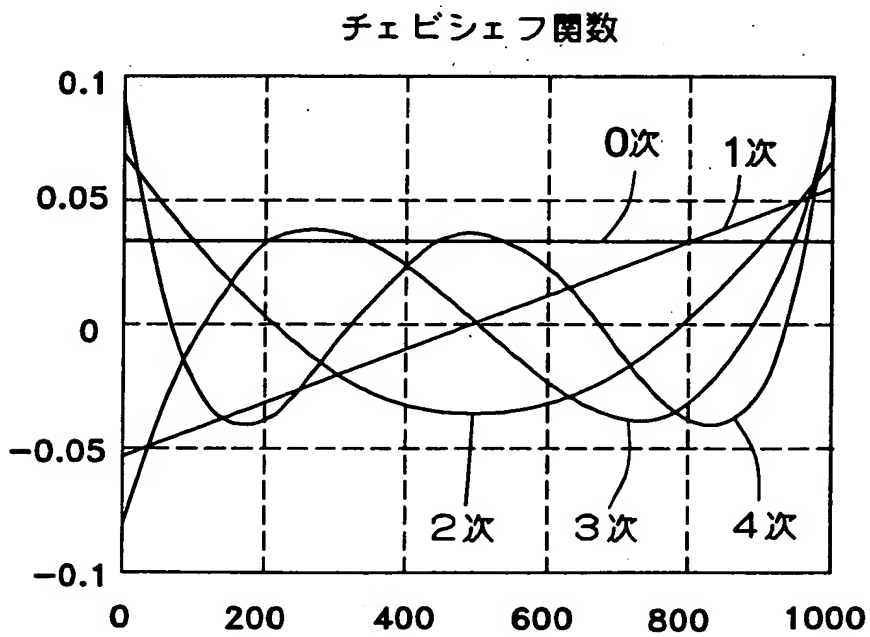
【図 2】



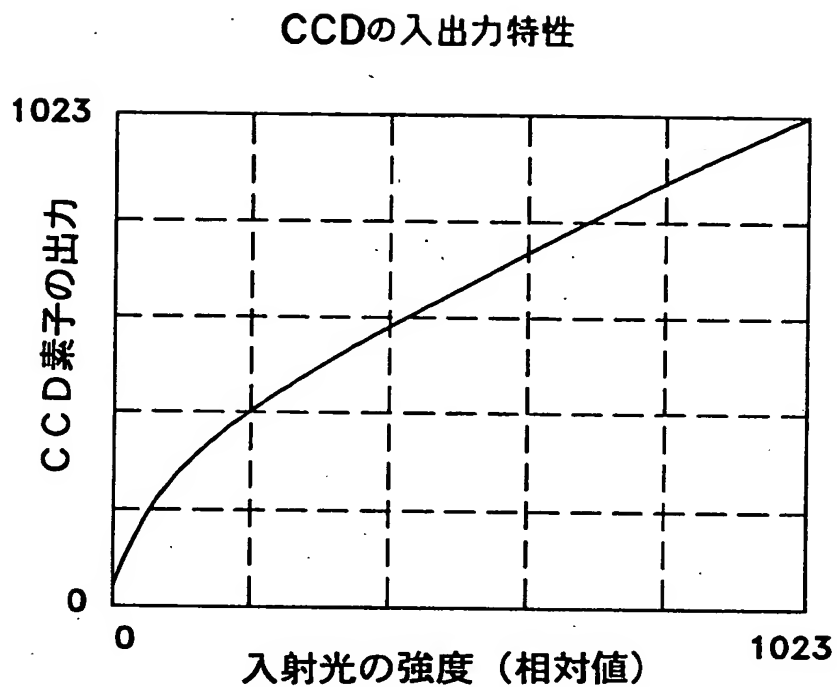
【図 3】



【図 4】



【図 5】





【書類名】 要約書

【要約】

【課題】 本発明の目的は、CCDが取り込んだデータを処理する際に、メモリ使用量及び演算量を抑えたデータ補正装置及びデータ補正方法を提供することにある。

【解決手段】 本発明のデータ補正装置10は、入力データの補正を行うための高次多項式を、直交展開するための展開係数及び基底関数のそれぞれの演算結果である展開係数配列データ32及び基底関数配列データ36が記憶された演算用メモリ22と、展開係数配列データ32及び基底関数配列データ36を利用して入力データの補正を行う演算部20と、を含む。高次多項式を直交展開したため、演算用メモリ22の使用量を抑えることができる。また、演算部20の演算時間を抑えることもできる。

【選択図】 図1

認定・付加情報

特許出願の番号	特願2001-064713
受付番号	50100327808
書類名	特許願
担当官	濱谷 よし子 1614
作成日	平成13年 4月 3日

<認定情報・付加情報>

【提出日】 平成13年 3月 8日

【特許出願人】

【識別番号】 390009531

【住所又は居所】 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】 100086243

【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内

【氏名又は名称】 坂口 博

【代理人】

【識別番号】 100091568

【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内

【氏名又は名称】 市位 嘉宏

【代理人】

【識別番号】 100106699

【住所又は居所】 神奈川県大和市下鶴間1623番14 日本アイ・ビー・エム株式会社大和事業所内

【氏名又は名称】 渡部 弘道

【復代理人】

【識別番号】 100094248

【住所又は居所】 滋賀県大津市栗津町4番7号 近江鉄道ビル5F  
楠本特許事務所

【氏名又は名称】 楠本 高義

次頁無

【書類名】 手続補正書

【あて先】 特許庁長官殿

【事件の表示】

【出願番号】 特願2001- 64713

【補正をする者】

【識別番号】 390009531

【住所又は居所】 アメリカ合衆国 1 0 5 0 4、ニューヨーク州アーモンク  
(番地なし)

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレ  
ーション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口博

【電話番号】 0462-73-3318

【復代理人】

【識別番号】 100094248

【住所又は居所】 滋賀県大津市粟津町 4 番 7 号近江鉄道ビル 5 F 楠本特許  
事務所

【弁理士】

【氏名又は名称】 楠本高義

【電話番号】 077-533-3689

【手続補正 1】

【補正対象書類名】 特許願

【補正対象項目名】 代理人

【補正方法】 変更

【補正の内容】

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口博

【電話番号】 0462-73-3318

【代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位嘉宏

【代理人】

【識別番号】 100106699

【弁理士】

【氏名又は名称】 渡部弘道

【その他】 代理人渡部弘道の氏名に誤記がありましたので訂正いたしました。

【ブルーフの要否】 要

認定・付加情報

特許出願の番号	特願2001-064713
受付番号	50100381681
書類名	手続補正書
担当官	濱谷 よし子 1614
作成日	平成13年 4月 3日

<認定情報・付加情報>

【提出日】	平成13年 3月16日
【補正をする者】	
【識別番号】	390009531
【住所又は居所】	アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
【氏名又は名称】	インターナショナル・ビジネス・マシーンズ・コーポレーション
【代理人】	
【識別番号】	100086243
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	坂口 博
【復代理人】	申請人
【識別番号】	100094248
【住所又は居所】	滋賀県大津市栗津町4番7号 近江鉄道ビル5F 楠本特許事務所
【氏名又は名称】	楠本 高義

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日	2000年 5月16日
[変更理由]	名称変更
住 所	アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
氏 名	インターナショナル・ビジネス・マシーンズ・コーポレーション